REMARKS

Claims 1-18 are pending in this application. By this Reply, claims 1 and 8 are amended. Reconsideration and withdrawal of the rejections are respectfully requested in view the foregoing amendments and the following remarks.

Claims 1-3, 5-11, and 13-18 stand rejected under 35 U.S.C. § 102(e) over Bentley et al. (U.S. Patent No. 5,815,415) (hereinafter Bentley). This rejection is respectfully traversed.

Claims 1, 8, and 15 broadly recite features of the preferred embodiment. Additionally, claims 1 and 8 have been amended to more particularly recite features of the preferred embodiment.

Bentley fails to disclose all the claimed features, as required by Section 102. For example, Bentley fails to disclose a management system kernel that provides management systems with a run-time environment, and a managed object generation environment that provides a development environment for managing applications, wherein the management system kernel can at least one of dynamically add and dynamically modify managed object information based upon an external meta file from the managed object generation environment without interrupting and operation of the network management system, as recited in claim 1 as amended.

Moreover, Bentley fails to disclose a network management method including, *inter alia*, storing a dynamic class loading routine in a management system kernel, and updating managed object information on the management system kernel without interrupting an operation of the management system by ... loading a dynamic library to the managed object framework utilizing the dynamic class loading routine, as recited in claim 8 as amended.

Finally, Bentley fails to disclose a network management method including storing a dynamic class loading routine in a management system kernel of the managed system, and updating the management system kernel by modifying managed object information in the management system kernel while a managed system is operating by utilizing the dynamic class loading routine, as recited in claim 15.

As is known in the art, a kernel is the essential center of a computer operating system. It is the core that provides basic services for all other parts of the operating system. Additionally, as discussed in the specification, the related art system method requires that an on-going network management system be stopped before a managed object can be added or modified. *See* page 5, lines 8-11.

Bentley relates to a computer aided design system, generally referred to as a computerized modeling system (CMS). Bentley discloses that the architecture of the CMS includes various layers, including a static kernel 12, a dynamic framework 14, and a portable persistent model 16. The kernel 12 provides the services necessary to load and execute the higher levels.

Bentley further discloses that the kernel 12 includes an object manager 30 that provides for the allocation, references, and persistence of all objects in the portable persistent model 16. Thus, whenever a new object is created, the object/persistence manager 30 creates an object descriptor for the object. See column 8, lines 18-28.

Referring next to column 8, lines 4-7, Bentley discloses that the kernel 12 can be extended by loading additional dynamic modules 23 with associated DLS files. The DLS files are Dynamic Link Specification files, and are provided with the kernel 12. Accordingly, they are not external to the kernel 12. The DLS files are used resolve the addresses of the native code functions in the kernel 12.

There is no disclosure in Bentley, however, of dynamically modifying managed object information based upon an external meta file from the managed object generation environment without interrupting an operation of the network management system.

Although at column 16, lines 47-55, Bentley discloses a "meta-class," the Bentley meta-class merely contains a description of an associated class object. Accordingly, there is no disclosure that the meta-class is used by the kernel to dynamically add and dynamically modify managed object information without interrupting and operation of the network management system.

Consequently, for at least these reasons, it is respectfully submitted that Bentley fails to disclose all of the claimed features, as required by Section 102. Claims 2, 3, and 5-7 depend from claim 1, claims 9-11, 13, and 14 depend from claim 8, and claims 16-18 depend from claim 15, and are allowable for at least the reasons discussed above with respect to the corresponding independent claims. Withdrawal of this rejection is thus respectfully requested.

Claim 4 stands rejected under 35 U.S.C. § 103(a) over Bentley in view Applicant's related art. This rejection is respectfully traversed.

Claim 4 depends from claim 1. As discussed above, Bentley fails to teach or suggest all the features of claim 1. Moreover, the recited portion of Applicant's background art (specification page 2, lines 5-6 and 10-18) fails to teach or suggest the features that are neither taught nor suggested by Bentley. Consequently, it is respectfully submitted that a prima facie case of obviousness cannot be made. Withdrawal of this rejection is thus respectfully requested.

Claim 12 stands rejected under 35 U.S.C. § 103(a) over Bentley, in view of Applicants related art, and further in view of Sheard et al. (U.S. Patent No. 6,208,345) (hereinafter Sheard). This rejection is respectfully traversed.

Claim 12 depends from claim 8. As discussed above, Bentley fails to teach or suggest all the features of claim 8. Moreover, neither the recited portion of Applicant's background art nor Sheard teaches or suggests the features that are neither taught nor suggested by Bentley. Moreover, the Patent Office does not rely on these references to teach such features. Consequently, it is respectfully submitted that a prima facie case of obviousness cannot be made. Withdrawal of this rejection is thus respectfully requested.

CONCLUSION

In view of the foregoing amendments and remarks, it is respectfully submitted that the application is in condition for allowance. If the Examiner believes that any additional changes would place the application in better condition for allowance, the Examiner is invited to contact the undersigned attorney, **Anthony H. Nourse**, at the telephone number listed below.

To the extent necessary, a petition for an extension of time under 37 C.F.R. 1.136 is hereby made. Please charge any shortage in fees due in connection with the filing of this, concurrent and future replies, including extension of time fees, to Deposit Account 16-0607 and please credit any excess fees to such deposit account.

Respectfully submitted,
FLESHNER & KIM, LLP

Daniel Y.J. Kim
Registration No. 36,186
Anthony H. Nourse
Registration No. 46,121

P.O. Box 221200
Chantilly, VA 20153-1200
703 502-9440
Date: November 25, 2002
AHN/DYK:ms

Amended Claims With Mark-ups to Show Changes Made

1. (Amended) A network management system, comprising:

a management system kernel that provides management systems with a run-time environment; and

a managed object generation environment that provides a development environment for managing applications, wherein the management system kernel can at least one of dynamically add and dynamically modify managed object (MO) information based upon an external meta file (EMM) from the managed object generation environment without interrupting and operation of the network management system.

8. (Amended) A network management method comprising:

(a) storing a dynamic class loading routine in a management system kernel;

(b) initializing a managed system by constructing a managed object framework of the management system kernel that contains information of managed object (MO) classes;

(c) creating MO instances and registering the MO instances in a containment tree of the management system kernel according to the information of MO classes;

(d) checking whether a dynamic class loading flag is on when receiving a management operation request from a management system; and

(e) updating MO information on the management system kernel without interrupting and operation of the management system by, waiting for all threads to complete execution,

loading a dynamic library to the managed object framework utilizing the dynamic class loading routine when the dynamic class loading flag is on, and

resetting the dynamic class loading flag to off.